

## 新人プログラマの生産性成長モデルとその評価

青山 幹雄 (富士通株式会社)

### 1. まえがき

ソフトウェア開発過程を管理するためには、開発状況を正確に把握する必要がある。

(1)~(3)

これまで、ソフトウェア生産性についての種々の計測が行なわれている。しかし、それらは何れもプロジェクト単位等の大まかなデータである。さらに、ソフトウェア開発効率の推移(変化)についての計測例はあまり無い。

一方、最近のソフトウェア需要の増大に伴い、多数の新人プログラマが、ソフトウェア開発に従事している。したがって、新人プログラマの生産性の把握がソフトウェア開発の管理上重要な問題となっている。

このような背景の下に、著者の従事した電子交換機(ESSと略記)のソフトウェア開発においては、プログラムの生産性等を詳細に測定した。本稿では、この測定データの分析から新人プログラマの生産性の推移が「成長曲線」でモデル化されることを明らかにした。

さらに、各モデルの近似度を評価し、生産性推移の要因を検討した。

### 2. 生産性の計測

#### 2-1 工程の定義

ESSのソフトウェア開発では、図2-1に示す「横割開発体制」を採った。

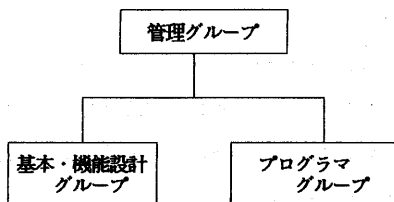


図2-1 横割開発体制

すなわち、基本～機能設計と詳細設計～プログラミングとを独立したグループが行なう工程別分業体制である。

計測対象の工程は、詳細設計からシミュレーションによる単体試験までである。以下、これをプログラム設計と呼ぶ。この工程における作業手順を、図2-2に示す。

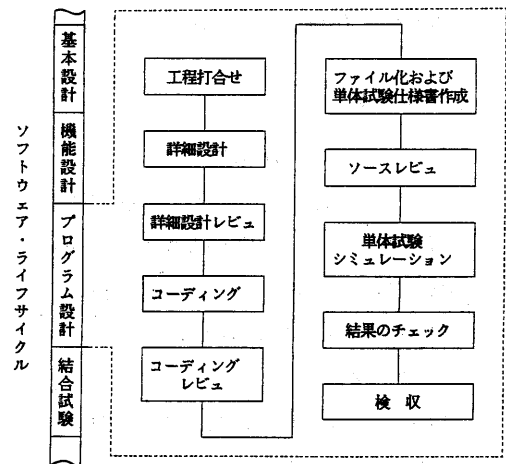


図2-2 プログラム設計工程の手順

#### 2-2 生産性の定義

プログラム生産性 $\eta$ を式(1)によって定義する。

$$\eta = \frac{\text{作成プログラムのソースステップ数}}{\text{該当プログラム作成の全工数}} \quad (1)$$

[ステップ/人日]

- (注) (1) ソースステップ数は、注釈行数を含まない。  
(2) プログラム記述言語としては構造化プログラム可能な高級言語「FSL (Fujitsu System Language for ESS)」を使用した。

\* これによってプログラム開発の生産性が前工程の生産性とは独立に測定できる。

データ定義、サブルーチンコール等アセンブラ  
でしか記述できない命令のみアセンブラで記述。

(3) 機能設計及び詳細設計のドキュメントには、  
「HIPO」を用いた。

### 2-3 測定方法

ソースステップ、工数の集計は開発単位 (モジュール)

毎に、各プログラマが集計した。

さらに、工数の集計は開発過程において、「詳細設計日  
程表」を用いて、日単位に行なった。

これらのデータは、グループリーダーのチェックを経  
て、検収時に、「開発報告書」として報告する体制にな  
っている。

### 2-4 サンプルング方法

ESSのソフトウェア開発に携った新人プログラマの  
中から、次の条件を満たす個人のデータを抽出した。

(1) 対象期間 (ソフトウェア開発に開始から5ヶ月間)

に継続して、かつ単独でプログラム開発に従事。

(2) 入社以前のソフトウェア経験が偏っていない。

### 2-5 データ処理

(1) 複数モジュールを並行 (同時) 開発した場合、その  
生産性として、10日間単位に式 (2) の荷重平均値  
を採用する。

$$\eta = \frac{\sum_{i=1}^m \eta_i \cdot W_i}{\sum_{i=1}^m W_i} = \frac{\sum_{i=1}^m S_i}{\sum_{i=1}^m W_i} \quad (2)$$

m: 並行開発のモジュール数、 $\eta_i$ : 各モジュールの生  
産性、 $S_i$ : 各モジュールのステップ数、 $W_i$ : 各モジ  
ュールの開発工数

(2) グループ化

新人教育においてプログラミング演習を長期間行な  
った新人と短期間のみ新人 (後述) とがあり、それ  
ぞれ、グループ I、II としてまとめた。

(3) 線型補間

測定データから生産性成長モデルを推定するために

次の局所的近似を施す。

① 各モジュールの生産性をその中点 (開発開始から  
終了までの中間時点) で代表する。

② 各モジュール間の生産性は、線型に変化すると仮  
定する。

### 3. 測定結果

測定値を個人毎に集計した結果を図3-1に示す。

さらに、2-5のデータ処理を施し各グループ毎に10日  
単位で平均値を求めた。

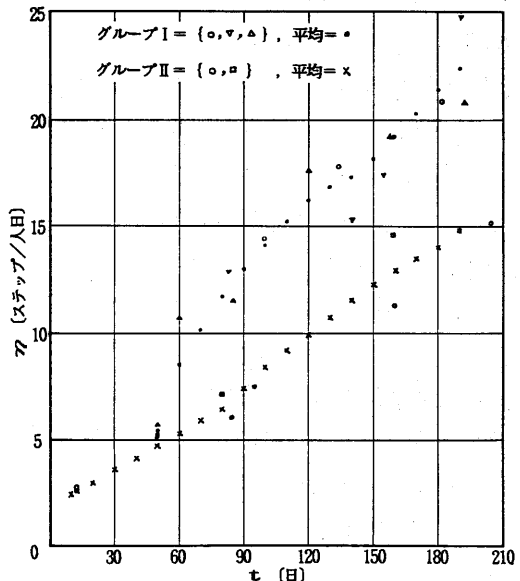


図3-1 生産性の測定結果

### 4. 成長曲線によるモデル化

人口の増加や経済変動の過程を表すモデルとして

Gompertz曲線、Logistic曲線等の成長曲線が知られている。

また、最近ではバグの累積過程を成長曲線を用いて評価し  
(5)(6) た例がある。

では、プログラム生産性の成長過程はどうか?

以下、生産性の測定データに対して成長曲線によるモデル  
化を行なった結果を示す。参考として、線型回帰分析によ  
るモデル化の結果も示す。

(7)  
数値計算には時系列分析ライブラリ「TIMS」を使用。

\* サンプルモジュールの平均ステップ数は、189行。

\*\* 本稿の調査対象者は全て理工系大学卒男子である。

4-1 成長曲線の諸性質

表4-1に示す性質がある。ただし、漸近線の80%に達するまでの期間を「成長期間」と定義した。

図4-1は成長曲線の概形を表す。これらの曲線には

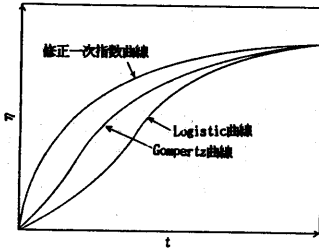


図 4-1 成長曲線の概形

	修正一次指数曲線	Gompertz 曲線	Logistic 曲線
定義式	$\eta_e = Ke - C \cdot d^t$ *	$\eta_g = Kg \cdot Q^t$	$\eta_l = \frac{Kl}{1 + m \exp(-at)}$
微分方程式	$\frac{\eta_e}{Ke} = \frac{1}{C \cdot \ln d} (\frac{Ke}{\eta_e} - 1)$	$\frac{\eta_g}{Kg} = (\ln t) (\ln \frac{\eta_g}{Kg})$	$\frac{\eta_l}{Kl} = a (1 - \frac{\eta_l}{Kl})$
変曲点	なし	$\bar{t} = (-\ln(-\ln a)) / \ln b$ $\bar{\eta} = Kg / e$	$\bar{t} = (\ln m) / \alpha$ $\bar{\eta} = Kl / 2$
成長期間	$\frac{\ln (Ke/5c)}{\ln d}$	$\frac{\ln (\ln 0.8 / \ln a)}{\ln b}$	$\frac{\ln (4m)}{\alpha}$
漸近線	$\eta_e^* = Ke$	$\eta_g^* = Kg$	$\eta_l^* = Kl$

表 4-1 成長曲線の諸性質

4-2 近似の評価

成長曲線の近似の良さを評価する指標として、次の諸量を

誤差率 (%)	不一致係数	相関係数	残差
$\frac{\eta_i - f_i}{f_i} \times 100$	$\frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (\eta_i - f_i)^2}}{\sqrt{\frac{1}{n} \sum_{i=1}^n \eta_i^2 + \frac{1}{n} \sum_{i=1}^n f_i^2}}$	$\frac{\sum_{i=1}^n f_i \cdot \eta_i}{\sqrt{\sum_{i=1}^n \eta_i^2 \cdot \sum_{i=1}^n f_i^2}}$	$\sqrt{\frac{1}{n} \sum_{i=1}^n (\eta_i - f_i)^2}$

用いた。

ただし、n：観測データ数、 $\eta_i$ ：観測値、 $f_i$ ：成長曲線による推定値 とする。

表 4-2 近似の評価式

4-3 モデル化の結果

図3-1の測定データに対して成長曲線のあてはめを行なった結果を図4-2に示す。図4-3は誤差率の推移を表す。表4-3は成長曲線のパラメータの推定値である。さらに、図4-4は近似の評価を示す。

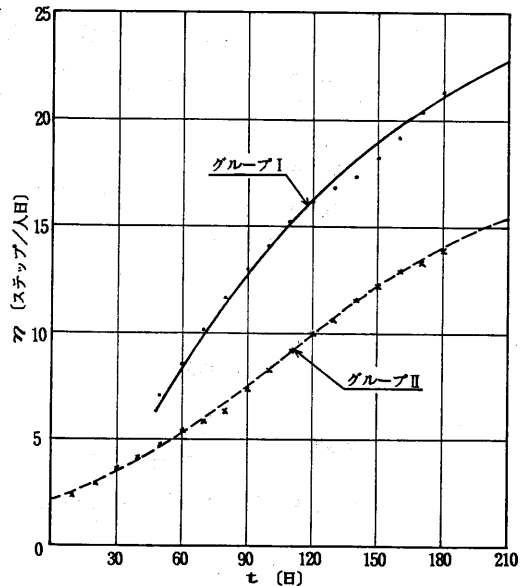


図 4-2 成長曲線のあてはめ

\* 修正一次指数曲線は次式の表現と等価である。

$$\eta = Ke - C \cdot e^{-\lambda t} = Ke (1 - r \cdot e^{-\lambda t})$$

$$\lambda = -\ln d, r = C/Ke$$

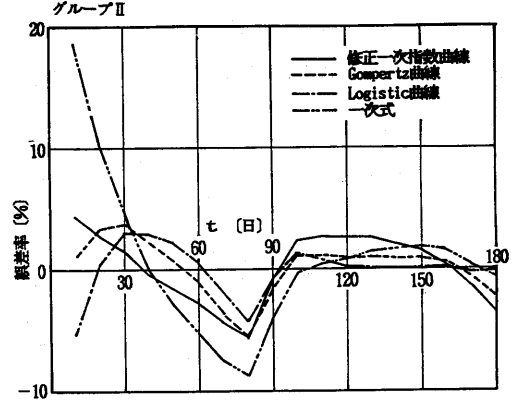
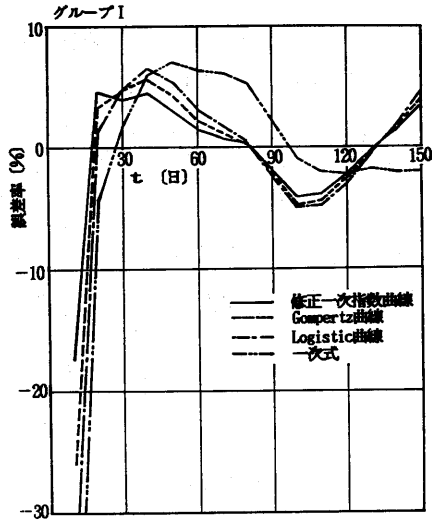


図 4-3 誤差率の推移

(グループ I)

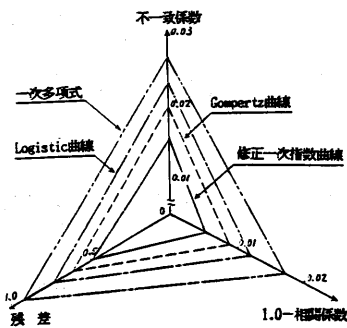
	修正一次指数曲線	Gompertz 曲線	Logistic 曲線
定義式	$\eta_e = 28.6 - 24.2 \cdot 0.921^{0.1x}$	$\eta_g = 24.8 - 0.221 \cdot 0.855^{0.1x}$	$\eta_L = \frac{23.3}{1 + 2.32 \exp(-0.0230x)}$
変曲点	なし	$\bar{t} = 26.3 \quad \bar{\eta}_g = 9.12$	$\bar{t} = 45.2 \quad \bar{\eta}_L = 11.7$
成長期間	174.8 日	121.7 日	105.5 日
漸近線	$\eta_e^\infty = 28.6$	$\eta_g^\infty = 24.8$	$\eta_L^\infty = 23.3$

(グループ II)

	修正一次指数曲線	Gompertz 曲線	Logistic 曲線
定義式	$\eta_e = -22.6 + 24.4 \cdot 1.024^{0.1x}$	$\eta_g = 25.2 - 0.0778 \cdot 0.920^{0.1x}$	$\eta_L = \frac{17.9}{1 + 7.188 \exp(-0.0182x)}$
変曲点	なし	$\bar{t} = 112.6 \quad \bar{\eta}_g = 9.27$	$\bar{t} = 108.1 \quad \bar{\eta}_L = 8.95$
成長期間	—	292.7 日	184.1 日
漸近線	なし	$\eta_g^\infty = 25.2$	$\eta_L^\infty = 17.9$

表 4-3 パラメータの推定値

グループ I



グループ II

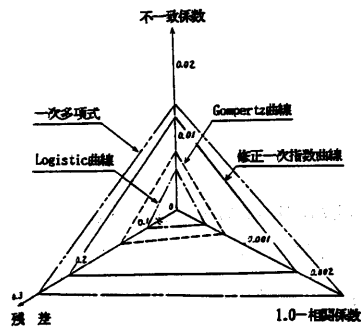


図 4-4 近似の評価

## 5. 考察

### 5-1 生産性の成長モデル

新人プログラムのソフトウェア生産性が、平均として成長曲線に従って向上することが明らかとなった。

何れのグループでも、局所的には線型補間を施したにも拘わらず、一次式より成長曲線が良い近似を得た。

さらに、グループによって異なる成長パターンを示している。すなわち、図4-4から近似の良さは次の順序であった。

グループI : 修正一次指数曲線 > Gompertz曲線  
> Logistic曲線 > 一次式

グループII : Logistic曲線 > Gompertz曲線  
> 修正一次指数曲線 > 一次式

この結果は、新人の生産性成長モデルが単一でないことを示している。

これらのモデルは、分散分析により危険率1%でその有意性を確認している。

また、グループIIのパラメータはモデル間の差が大きい。測定データが飽和領域に達していないため、Logistic曲線とGompertz曲線の変曲点に関する対称、非対称の差によるものである。

さらに、測定期間以後の大域的な成長モデルが問題となる。グループIには、積極的-消極的加速度曲線に近い成長例が見られた。

### 5-2 導入教育の効果

McGeoch は練習効果の向上過程を図5-1に示す三種に分類した。

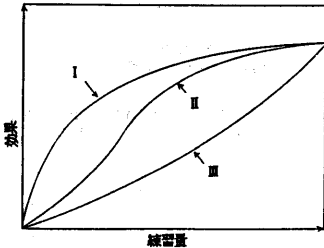


図 5-1 練習曲線の型

成長曲線とMcGeoch の「練習曲線」とを比較すれば、

次の対応関係がある。

型	対応するモデル
I	修正一次指数曲線 ( $C > 0$ )
II	Gompertz曲線, Logistic曲線
III	修正一次指数曲線 ( $C < 0$ )

以前に類似の学習経験がある場合I型の練習曲線となる<sup>(9)</sup>と言われている。この場合は、グループIの新人が、教育(1.5カ月間)においてソフトウェア開発の演習を行なった効果であろう。

### 5-3 総開発量の評価

図5-2は教育期間(40日)を考慮しグループI、IIの一人当たりの総開発量( $\sum_{i=1}^n \eta_i \cdot 10$ )をモデルから推定した結果を示す。教育期間による開発開始の遅れを考慮しても、2ヶ月後には、グループIの総開発量が上まわっている。

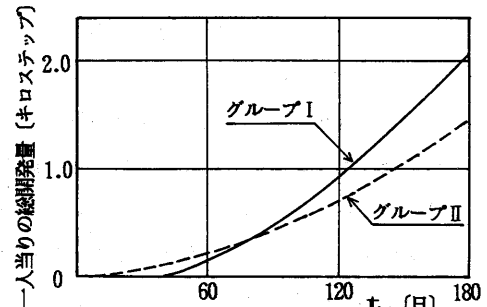


図 5-2 総開発量の推移

### 5-4 作業標準化の効果

ESSのソフトウェア開発では、本稿で対象とした開発作業に先行して、生産性の試作調査を行った。

この調査では、(1)9名の新人に、(2)三回にわたり、(3)異なるプログラムの作成を課して、生産性を測定した。

図5-3に代表例を示す。同図からは生産性の向上が明確でない。この理由は、主に次の三点にあった。

#### (1) プログラム記述レベルの不統一

FSLはアセンブラの混合記述が可能であるため、その混合比により同一機能のモジュールであっても見かけのソースステップ数が異なる。

(2) 作業方法の不統一

作業の標準化が図られていなかったため、個人差が増大した。

例 ドキュメントの記述、アSEMBル等のJCL

(3) 生産性の測定・管理の不統一

測定基準、管理体制が確立されていなかった。

この試行結果に基づいて、ESSのソフトウェア開発では詳細な作業標準を作成・実施した。それによって、開発状況の正確な把握ができ、開発管理をするうえで有益であった。

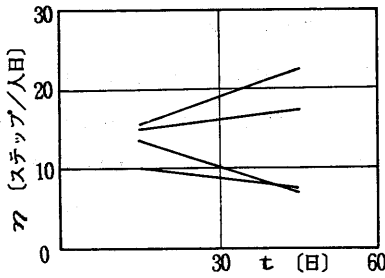


図 5-3 試行段階の生産性

5-5 個人差について

各グループ内の生産性のばらつきの推移を図5-4に示す。他の生産性に関する報告に比べばらつきが少ない。これは、(1)サンプリング条件によって経験等のばらつきの要因が少なくなった、(2)作業標準の徹底により作業の「自由度」が少なくなったことによると思う。

今後、2-4のサンプリング条件を満たさないデータの分析も必要と思う。

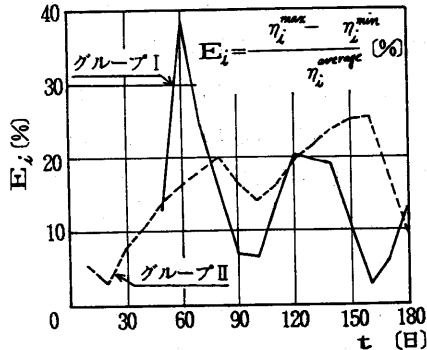


図5-4 ばらつきの推移

5-6 生産性向上のために

ソフトウェア開発過程において成長曲線を作成することにより、新人プログラマ自身が、(1)生産性向上の状況把握、(2)次に目標とすべき生産性の設定ができるため生産性向上の一ツールとして有用と思われる。

6. あとがき

ソフトウェア生産はその属人性の大きい故に正確な測定は困難である。しかし、ソフトウェア開発を実際に管理するためには生産性の把握が不可欠である。今後、多数の生産性データの収集と分析によってその理解を深め、生産性向上の問題点を検討したい。

7. 謝 辞

日頃御指導いただき、当社通信ソフトウェア事業部の菅原課長、蓮井浩哉部長はじめ関係各位に感謝します。

8. 参考文献

- (1) B.W.Boehm : "The high cost of software" Software World Vol.6, No.1, pp.2-10
- (2) T.C.Jones : "Measuring programming quality and productivity" IBM Syst.J., Vol.17, No.1, pp.39-63
- (3) K.Christensen et.al. : "A perspective on software science" IBM Syst.J., Vol.20, No.4, pp.372-383
- (4) 蓮井浩哉 他 : "電子交換システム開発用語集 FSL" FUJITSU Vol.31, No.1, pp31-37
- (5) 三鶯 武 : "ソフトウェアの品質評価" 日科技連 1981
- (6) 山田 茂 他 : "ソフトウェアエラー発見過程の確率モデルとその応用" 情報処理学会ソフトウェア工学研究会資料, 21-3 1981
- (7) "FACOM TIMS 解説書" FACOM マニュアル
- (8) 辰野千壽 : "学習心理学総説" 金子書房 1973
- (9) 木村正男 他 : "心理学概論" 福村出版 1980
- (10) B.W.Boehm : "Software Engineering Economics" Prentice - Hall, 1981